untitled14

April 4, 2025

[8]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[24]:

```
dt = sns.load_dataset('titanic')
dt.head()
```

[24]:

```
   survived  pclass     sex   age  sibsp  parch     fare embarked  class  \
0         0       3    male  22.0      1      0   7.2500        S  Third
1         1       1  female  38.0      1      0  71.2833        C  First
2         1       3  female  26.0      0      0   7.9250        S  Third
3         1       1  female  35.0      1      0  53.1000        S  First
4         0       3    male  35.0      0      0   8.0500        S  Third

     who  adult_male deck  embark_town alive  alone
0    man        True  NaN  Southampton    no  False
1  woman       False    C    Cherbourg   yes  False
2  woman       False  NaN  Southampton   yes   True
3  woman       False    C  Southampton   yes  False
4    man        True  NaN  Southampton    no   True
```

[29]:

```
dt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   survived     891 non-null    int64
 1   pclass       891 non-null    int64
 2   sex          891 non-null    object
 3   age          714 non-null    float64
 4   sibsp        891 non-null    int64
 5   parch        891 non-null    int64
 6   fare         891 non-null    float64
 7   embarked     889 non-null    object
 8   class        891 non-null    category
 9   who          891 non-null    object
 10  adult_male   891 non-null    bool
 11  deck         203 non-null    category
 12  embark_town  889 non-null    object
 13  alive        891 non-null    object
 14  alone        891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.7+ KB
```

[30]:

```python
dt.describe()
```

[30]:

```
         survived      pclass         age       sibsp       parch        fare
count  891.000000  891.000000  714.000000  891.000000  891.000000  891.000000
mean     0.383838    2.308642   29.699118    0.523008    0.381594   32.204208
std      0.486592    0.836071   14.526497    1.102743    0.806057   49.693429
min      0.000000    1.000000    0.420000    0.000000    0.000000    0.000000
25%      0.000000    2.000000   20.125000    0.000000    0.000000    7.910400
50%      0.000000    3.000000   28.000000    0.000000    0.000000   14.454200
75%      1.000000    3.000000   38.000000    1.000000    0.000000   31.000000
max      1.000000    3.000000   80.000000    8.000000    6.000000  512.329200
```

[27]:

```python
dt.isnull().sum()
```

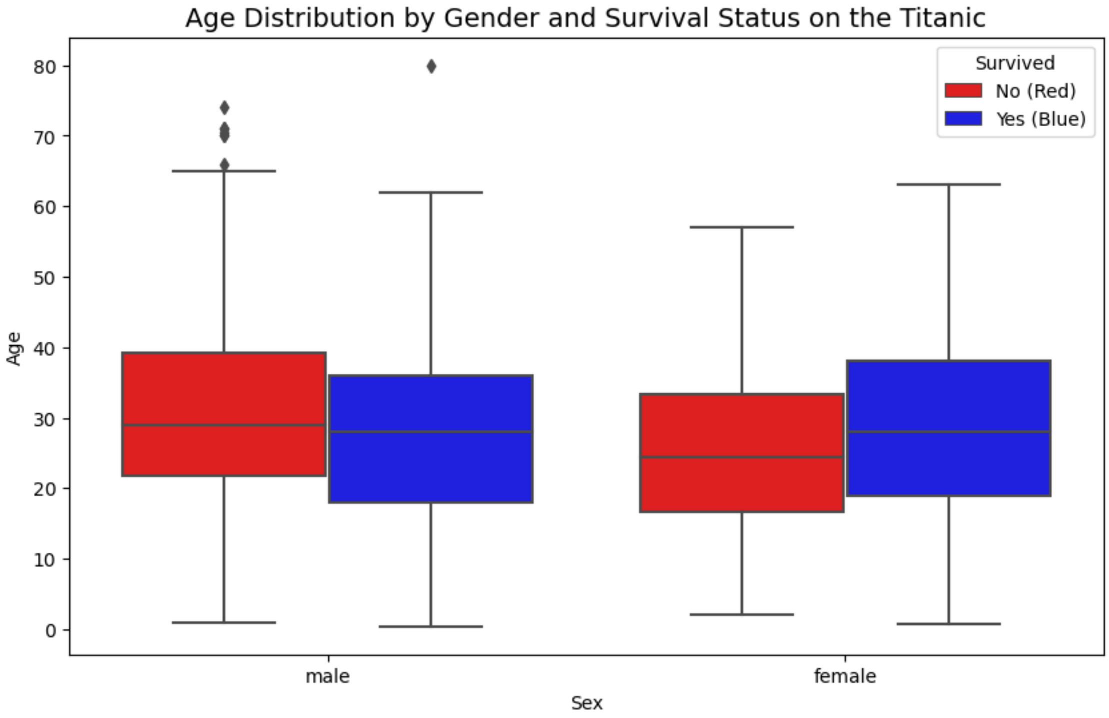
[27]:

```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

[16]:

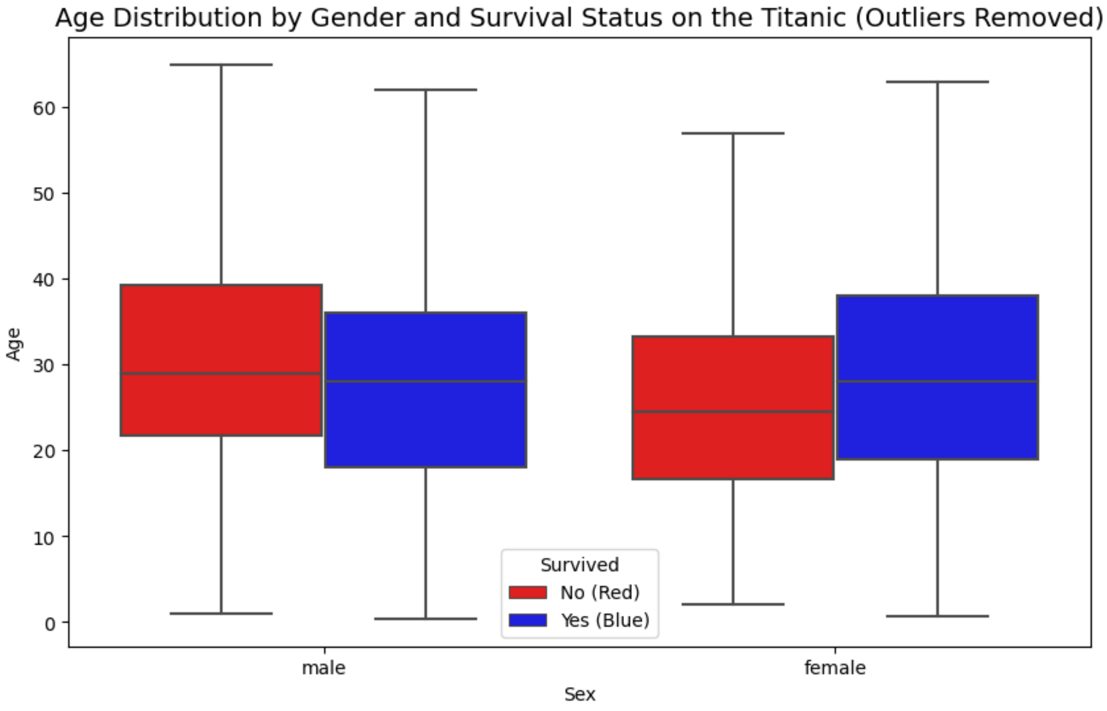
```python
import seaborn as sns
import matplotlib.pyplot as plt

# Load Titanic dataset
dt = sns.load_dataset('titanic')

# Define highly contrasting colors
```

```
custom_palette = {0: 'red', 1: 'blue'}  # 0 = Not Survived (Red), 1 = Survived␣
  ↪(Blue)

# Create the box plot
plt.figure(figsize=(10, 6))
sns.boxplot(x='sex', y='age', hue='survived', data=dt, palette=custom_palette)

# Fix legend labels properly
legend_labels = ['No (Red)', 'Yes (Blue)']
handles, _ = plt.gca().get_legend_handles_labels()
plt.legend(handles, legend_labels, title='Survived')

# Add title and labels
plt.title('Age Distribution by Gender and Survival Status on the Titanic',␣
  ↪fontsize=14)
plt.xlabel('Sex')
plt.ylabel('Age')

# Show plot
plt.show()
```



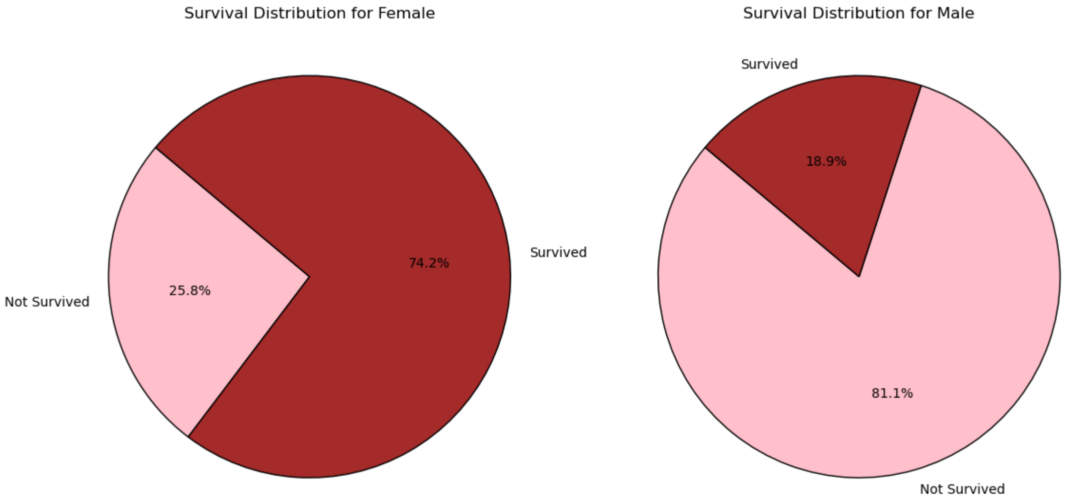
[22]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

```python
# Load Titanic dataset
dt = sns.load_dataset('titanic')

# Define distinct colors for survival status
custom_palette = {0: 'red', 1: 'blue'}  # 0 = Did not survive, 1 = Survived

# Create the box plot without outliers
plt.figure(figsize=(10, 6))
sns.boxplot(x='sex', y='age', hue='survived', data=dt, palette=custom_palette, showfliers=False)

# Fix legend labels properly
legend_labels = ['No (Red)', 'Yes (Blue)']
handles, _ = plt.gca().get_legend_handles_labels()
plt.legend(handles, legend_labels, title='Survived')

# Add title and labels
plt.title('Age Distribution by Gender and Survival Status on the Titanic (Outliers Removed)', fontsize=14)
plt.xlabel('Sex')
plt.ylabel('Age')

# Show plot
plt.show()
```



[20]:

```
# Count of survival status grouped by gender
survival_counts = dt.groupby(['sex', 'survived']).size().unstack()

# Define colors for better visualization
colors = ['pink', 'brown']  # Red = Not survived, Blue = Survived

# Create the pie charts
fig, axes = plt.subplots(1, 2, figsize=(12, 6))

for i, gender in enumerate(survival_counts.index):
    axes[i].pie(
        survival_counts.loc[gender],
        labels=['Not Survived', 'Survived'],
        autopct='%1.1f%%',
        colors=colors,
        startangle=140,
        wedgeprops={'edgecolor': 'black'}
    )
    axes[i].set_title(f'Survival Distribution for {gender.capitalize()}')

# Show plot
plt.tight_layout()
plt.show()
```



[ ]: